ADMINISTRACIÓN DE BASE DE DATOS CON POSTGRESQL LABORATORIO 1. CONTROL DE USUARIOS

Luis Antonio Álvarez Oval

Universidad Autónoma de Chiapas

MTRO. LUIS ANTONIO ÁLVAREZ OVAL

Mtro. en Ciencias, PTC de la Fac. Contaduría
UNACH Campus Tapachula. Contacto loval@unach.mx

RESUMEN

La serie de laboratorios de Administración de Bases de Datos con PostgreSQL, muestra de forma práctica la administración de este tipo de sistemas, el cual tiene un amplio uso en la industria de desarrollo de software. Mientras que las bases de datos son la herramienta que requieren las empresas que necesitan almacenar la información que generan, es en este tipo de sistemas donde se guarda ésta. De ahí la importancia de entender y aplicar los conceptos de administración estándar que se usa en la industria. Se usa el sistema PostgreSQL debido a que ofrece los mecanismos que tienen otros sistemas similares pero de carácter propietario. PostgreSQL se ofrece bajo una licencia PostgresSQL, lo que permite desde el punto de vista del propietario de un sistema de información evitar el pago de costosas licencias por el uso de una base de datos

Palabras Claves: *Administración de Base de Datos, SQL, Programación de procedimientos almacenados, postgreSQL.*

Esta serie de seis laboratorios de Administración de Base de Datos (ABD) son un punto de partida para conocer a detalle las prestaciones ofrecidas por el sistema PostgreSQL, este es un sistema de base de datos objeto-relacional que tiene las características de los sistemas de base de datos propietarios tradicionales. PostgreSQL es libre y el código fuente completo está disponible. Esta última característica es la más atractiva para desarrollar aplicaciones empresariales para el mercado latinoamericano, ya que evita el pago de costosas licencias. El software y la documentación se ofrecen bajo la licencia PostgreSQL (http://www.postgresql.org/about/licence/), la cual es similar a las licencias BSD o MIT. Los laboratorios se han diseñado para proporcionar los conceptos y la experiencia necesarios para conocer detalladamente el sistema, se aprovecha la función de "copiar y pegar" que nos ofrece el sistema operativo Windows para disminuir el esfuerzo del lector en la preparación del ambiente de trabajo y en la solución de los problemas.

En la sección denominada "trabajo adicional" se requiere que el lector aplique la experiencia obtenida en la solución de problemas relacionados con el tema central del laboratorio. La sección de conceptos básicos muestra la sintaxis de los comandos y da algunas explicaciones del uso de los mismos. Este material ha sido tomado del *Manual de usuario del sistema PostgreSQL* el cual está disponible en la página oficial de la herramienta, en algunos casos se ha tomado del sitio oficial en Español. Los conceptos básicos se aplican en torno a un proyecto que se denomina "Universidad ACME", el cual es producto de la imaginación del autor, así como la solución práctica de los problemas planteados. Los libros que se ofrecen en la sección de referencias, sirven como consulta para apoyar algunos de los conceptos que se aplican en la solución práctica de problemas de administración de base de datos.

Estos laboratorios se han preparado para procurar una experiencia práctica a los estudiantes de la materia Administración de Base de Datos de la Licenciatura en Sistemas Computacionales que se ofrece en la Facultad de Contaduría Pública (FCP) del Campus IV de la Universidad Autónoma de Chiapas (UNACH). En la FCP tenemos al menos 14 años de experiencia en el uso de PostgreSQL en las aulas,

proyectos de investigación y en sistemas que se han implementado para la automatización de las actividades cotidianas de la FCP. Como producto de esa experiencia académica e industrial se han obtenido estos laboratorios que se usan en las aulas para capacitar a nuestros estudiantes. Hemos encontrado que los estudiantes se motivan al estudio cuando se concretan en estos ejercicios las ideas abstractas que se explican en las aulas, aunque ese será tema de otro artículo. También se tiene noticia de que son una fuente de consulta para egresados que laboran en el sector empresarial.

Como se ha mencionado previamente la herramienta tiene características y lenguajes de programación estándar que ofrecen sistemas propietarios, por lo que los ejemplos fácilmente pueden ser aplicados en otros sistemas de bases de datos del mercado, o pueden ser referencia para aplicar los conceptos en proyectos industriales. Por lo que puedan servir como consulta a profesionales de las Ciencias de la Computación.

OBJETIVO

El lector aprenderá a administrar grupos y usuarios que acceden a una base de datos, así como a otorgar y revocar privilegios para limitar sus actividades usando las herramientas que ofrece el sistema de administración de base de datos PostgreSQL.

PRERREQUISITOS

Se espera que el lector tenga experiencia previa en el uso y conversión de diagramas Entidad-Relación (E-R), los temas asociados al Diseño de Base de Datos no se cubren en este documento. También se espera que el usuario tenga conocimientos básicos del lenguaje de programación denominado SQL.

Es necesario instalar la base de datos PostgreSQL versión 9.3 sobre el sistema operativo Windows, verifique los requerimientos para instalación en la página oficial de la herramienta: www. postgresql.org.



El sistema puede descargarse del sitio Web: http://www.enterprisedb.com/products-services-training/pgdownload#windows

PARTES QUE COMPONEN ESTE LABORATORIO

1. Proyecto a desarrollar
2. Conceptos básicos
3. Preparación del ambiente de trabajo
4. Problemática a resolver
5. Trabajo adicional
6. Referencias

I. Proyecto a desarrollar

El ejercicio consiste en un proyecto que describe el problema de una empresa dedicada a la prestación de servicios educativos: después de leer el texto se genera el diagrama E-R con la solución a este problema, se continúa con la creación de las tablas y población de las tablas, para finalmente trabajar con los permisos de grupos y usuarios.

Proyecto universidad ACME
En UACME, se ofrecen dos tipos de cursos en el periodo especial de verano, en que se imparten cursos de verano y cursos extracurriculares. Los primeros son materias que un alumno regular que estudia una carrera cursa en este periodo, se le permite adelantar hasta dos materias; mientras que los segundos son cursos especiales de capacitación que se ofrecen a alumnos regulares como estudiantes o profesionistas externos.

Los docentes de la UACME, son los únicos a los que se les permite impartir estos cursos, por los cuales reciben un pago adicional, se les paga de acuerdo con un tabulador que indica el costo de la hora de estos cursos de acuerdo al nivel académico del docente. El pago se genera a partir del alta del curso y sólo se permite expedir un cheque por cada curso. Además, los estudiantes deben acudir a pagar adicionalmente al costo del semestre por asistir a ellos.

UACME tiene dos departamentos que intervienen en la administración de los cursos:

A) Departamento de Administración (DA) y B) Departamento de Control Escolar (DCE). Corresponde al DA, efectuar el pago a los docentes y los cobros a los alumnos. El DA es dirigido por el C.P. Ávila y es auxiliado por el Sr. Cancino. Mientras que el DCE, es dirigido por el Lic. Barroso y auxiliado por los Sras. Tirado, Martínez, Aquino y Ramos y es en éste donde se decide qué cursos se imparten en el periodo, quién los imparte, y se aceptan las solicitudes de los alumnos. Un caso especial, es el de los Profesores, ya que el DA es quién les puede modificar el sueldo quincenal, mientras que el DCE ni siquiera puede visualizar este. Lo curioso es que el DCE es quien acepta los docentes y los registra en el sistema, pero es el DA donde se captura el sueldo. Importante es para la administración de la UACME que esta política se aplique al pie de la letra, y que sea implementado directamente sobre la DB. A continuación se describe detalladamente las tablas a las cuales tiene acceso el personal.

Tablas a las que se le permite el acceso al personal de la Secretaría Administrativa: Cuenta Cheques, Cheque, Tabulador, Profesores, Concepto, Recibo, y Detalle Recibo.

Como casos especiales este departamento podrá acceder a consultar las tablas de Cursos Especiales, Cursos Especiales Verano, Cursos Especiales Extracurriculares, Cursos Extracurriculares y Materias. Explícitamente no se les permite modificar ningún campo o registro.

Tablas a las que se le permite el acceso al personal de la Secretaría Escolar: CursosEspeciales, CursosExtracurricular, Materias, CEVerano, CEExtracurricula, Alumnos, Bimestre, Faltas, CalendarioEscolar.

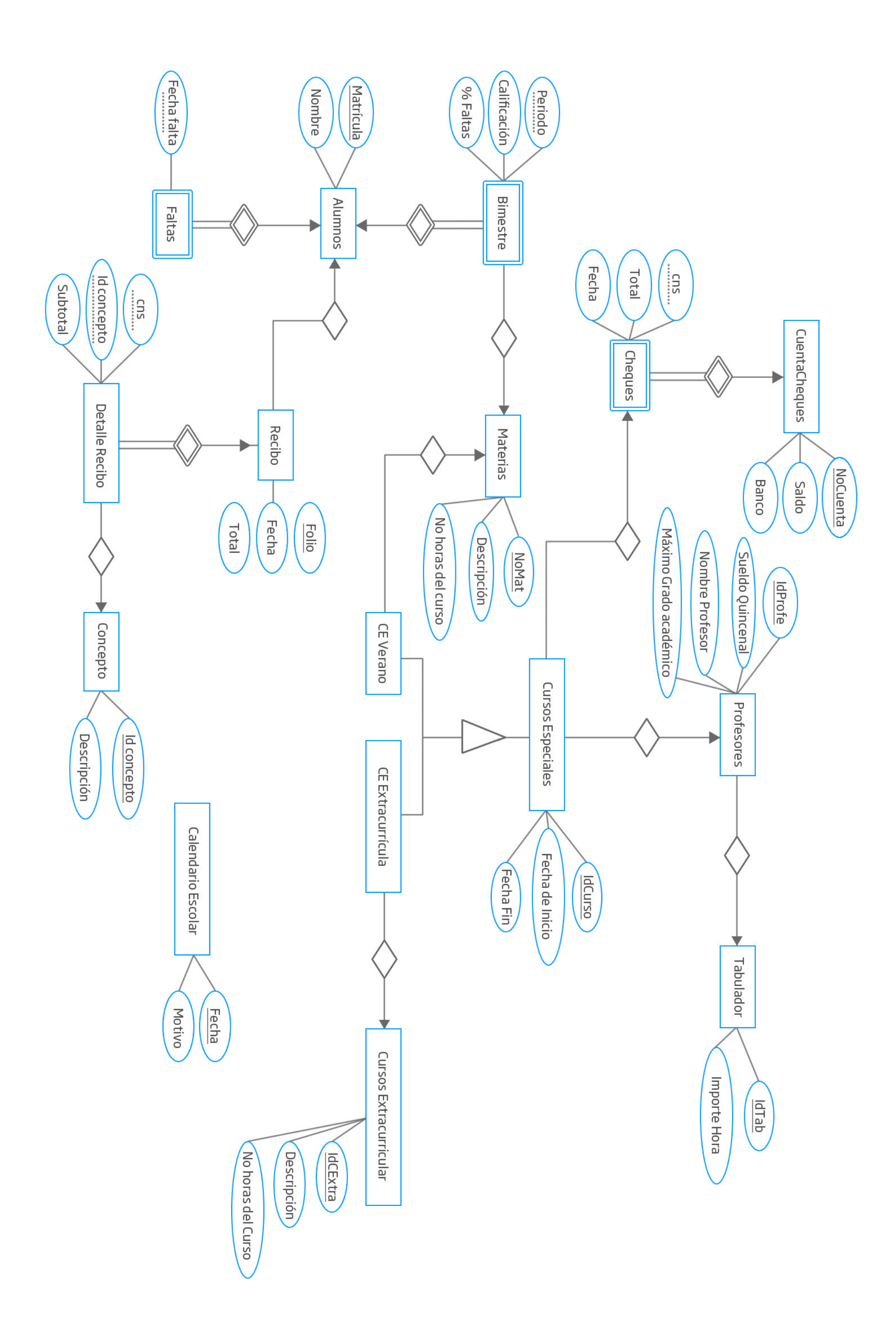


Figura 1. Diagrama E/R que resuelve el problema anterior

II. Conceptos básicos

Aquí encontrará una versión modificada del manual de usuario de PostgreSQL que da una explicación del uso y la sintaxis de los comandos usados en el presente laboratorio. Para consultar el manual oficial en idioma inglés visite el sitio en Internet.

Administrando Cuentas de Usuario
Como un administrador de PostgreSQL, quizá sea responsable de crear cuentas de usuarios y grupos. Quizá sea responsable también de otorgar y revocar privilegios.

En la mayor parte de los ambientes, hay un mapeo uno a uno entre la identidad de los usuarios en el sistema operativo y la identidad de PostgreSQL. En efecto, tu nombre de usuario en PostgreSQL es frecuentemente identificado con tu nombre de usuario del sistema operativo.

En algunos casos otras configuraciones son útiles. Por ejemplo, quizá quiera que la mayoría de los usuarios se identifique por ellos mismos de manera única mientras les proporciona una cuenta con privilegios de invitado. Quizá también tiene una aplicación cliente que se identifica por sí misma pero rara vez identifica al usuario (esto es útil para las aplicaciones que son ejecutadas por algún usuario dentro de algún proveedor de autentificación).

Una cuenta de usuario es compartida entre todas las bases de datos dentro de clúster dado. Los grupos de usuarios son también almacenados entre todas las bases de datos dentro de un clúster.

Privilegios (CREATEDB y CREATEUSR)
Cuando se crea un nuevo usuario, se puede controlar algunas de las actividades que estos realizan en la base de datos, tal como el permiso de crear nuevas bases de datos. También puede controlar la actividad de crear nuevos usuarios. Dar al usuario el derecho de crear nuevas bases de datos o crear nuevos usuarios es un riesgo. Cuando asigna privilegios a un usuario con CREATEUSER, ese usuario llega a ser un súper usuario en el clúster. Se debe decir que ésta es

una forma ligeramente diferente: un usuario que tiene privilegios por CREATEUSER puede sobrepasar todas las restricciones en el clúster de base de datos. Se puede negar explícitamente los privilegios con CREATEUSER especificando NOCREATEUSER, NOCREATEUSER es asumido si no se específica otro valor.

Las opciones de CREATEDB dan al usuario el derecho de crear nuevas bases de datos (dentro del clúster). Se puede especificar NOCREATEDB para prohibir al usuario crear nuevas bases de datos. Si especifica CREATEDB no NOCREATEDB, CREATE USER asumirá NOCREATEDB.

Administración de Grupos

Se pueden definir grupos de usuarios para hacer la administración mucho más fácil. Cada grupo puede incluir usuarios. Cada usuario puede llegar a pertenecer a uno o más grupos. Cuando se otorga o revocan privilegios para un objeto, se puede identificar un usuario específico o un grupo de usuarios.

Cada usuario es automáticamente miembro de un grupo PUBLIC. PUBLIC es realmente un grupo virtual, no puede agregar o remover miembros y no puede borrar este grupo, pero se permite asociar privilegios con PUBLIC.

Los grupos son mucho más fáciles de manejar, si ellos coinciden con los roles en la organización. Por ejemplo, quizá construya grupos nombrados como: desarrolladores, invitados, cajeros y administradores. Los grupos se deben ordenar de forma que reflejen el mundo real, los grupos hacen que sea mucho más fácil asignar privilegios en los objetos de la base de datos. Un objeto dado puede pertenecer a muchos grupos. Por ejemplo, un miembro del grupo de los desarrolladores quizá también pertenezca al de los administradores.

Las definiciones de los grupos son almacenados en las tablas del sistema. Como los usuarios de la base de datos, las definiciones del grupo son almacenadas para toda la base de datos dentro de un clúster.

CREATE GROUP

Un súper usuario PostgreSQL puede crear un nuevo grupo usando el comando CREATE GROUP:

```
CREATE GROUP nombre_grupo [ [WITH] option [….] ]
```

El nombre de grupo debe reunir las reglas para los identificadores de PostgreSQL (31 caracteres o menos, comillas o iniciar con una letra o subrayado). Puedes incluir un valor SYSID si quiere asignar un identificador numérico para el nuevo grupo. Nosotros conocemos cada grupo de usuarios por su nombre, pero alguna tabla que se refiera al grupo se referirá al valor numérico. Quizá asigne un identificador numérico específico para un grupo por las mismas razones que se asigna un identificador específico para un usuario.

Puede asignarse un miembro a un grupo de tres maneras:

- Usar la opción IN GROUP en el comando CREATE USER
- Lista los nombres de usuarios en el USER opción de CREATE GROUP
- Cambia los miembros del grupo usando el comando ALTER GROUP

Un típico comando es CREATE GROUP con el que quizá se vea algo como esto:

```
CREATE GROUP desarrolladores USER Bernardo, lety;
[ [WITH] option]…
```

Este comando crea un nuevo grupo nombrado desarrolladores que inicialmente tiene dos miembros Bernardo y Lety.

Crear Usuarios

Hay dos formas de crear un nuevo usuario: ejecutando el comando CREATE USER desde una aplicación cliente (tal como un psql), o con el createuser del shell script.

La sintaxis completa para el comando CREATE USER es:

```
CREATE USER nombre_usuario
[ [WITH] option]…

option := SYSID user-id-number
| [NO] CREATEDB
| [NO] CREATEUSER
| IN GROUP groupname[, …]
| [ [UN] ENCRYPTED ] PASSWORD 'password'
| VALID UNTIL 'expiration'
```

Un nombre-usuario debe conformar las reglas usuales para los identificadores PostgreSQL: esto debe iniciar con una letra (o un subrayado) y debe tener 31 caracteres de extensión. Si necesitas iniciar un nombre de usuario con un número, sólo encierra el nombre con comillas dobles.

Vistas

Las vistas son pseudo-tablas, esto es, que no son tablas reales, sin embargo aparecen como tablas ordinarias para seleccionar. Una vista puede representar un subconjunto de una tabla real, seleccionando ciertas columnas o ciertas filas de una tabla ordinaria. Incluso, una vista puede representar a varias tablas unidas. Debido a que a las vistas se les asignan permisos por separado, se les puede usar para restringir acceso a una tabla. Las vistas son creadas utilizando el comando CREATE VIEW.

Creación de reglas

Sintaxis

```
CREATE RULE name AS ON event
TO object [ WHERE condition ]
DO [ INSTEAD ] [ action | NOTHING ]
```

name - El nombre de la regla a crear.
event - Evento puede ser select, update, delete o insert.
object - Objeto puede ser *table* o *table.column*.

condition - Cualquier clausula, SQL WHERE, New o Current, pueden aparecer en lugar de una variable de instancia siempre que una variable de instancia es admisible en SQL.
action - Cualquier clausula SQL, New o Current pueden aparecer en lugar de una variable de instancia siempre que una variable de instancia sea admisible en SQL.

Descripción

El sistema de reglas de PostgreSQL permite que una acción alternativa sea realizada en *updates*, *inserts* o *deletes* en tablas o clases. Actualmente se utilizan reglas para implementar vistas de tablas.

El significado de una regla es que cuando una instancia individual es accedida, actualizada, insertada o borrada, existe una instancia actual (para consultas, actualizaciones y borrados) y una nueva instancia (para actualizaciones y añadidos). Si el *event* especificado en la cláusula ON y la *condition* especificada en la cláusula WHERE son verdaderas para la instancia actual, la parte *action* de la regla es ejecutada. Antes, sin embargo, los valores de los campos de la instancia actual y/o la nueva instancia son sustituidos por current.*attribute-name* y new.*attribute-name*.

La parte *action* de la regla se ejecuta con el mismo identificador de comando y transacción que el comando de usuario que causó la activación.

Notas

Es pertinente la precaución con reglas de SQL. Si el mismo nombre de clase o variable de instancia aparece en el event, la condition y la parte action de la regla, son considerados todos diferentes tuplas. De forma más precisa, new y current son las únicas tuplas que son compartidas entre cláusulas. Por ejemplo, las siguientes dos reglas tienen la misma semántica.

```
ON UPDATE TO emp.salary WHERE emp.name = "Joe"
DO UPDATE emp ( ... ) WHERE ...
```

```
ON UPDATE TO emp-1.salary WHERE emp-2.name = "Joe"
DO UPDATE emp-3 ( ... ) WHERE ...
```

Cada regla puede tener la etiqueta opcional INSTEAD. Sin esta etiqueta, la *action* será realizada en adición al comando de usuario cuando el *event* en la parte *condition* de la regla aparezcan. Alternativamente, la parte *action* será realizada en lugar del comando del usuario. En este último caso, la *action* puede ser la palabra clave NOTHING.

Cuando se elige entre los sistemas de reescritura y reglas de instancia para una aplicación particular de una regla, recuérdese que en el sistema de reescritura, *current* se refiere a la relación y algunos calificadores mientras que en el sistema de instancias se refiere a una instancia (tupla). Es muy importante notar que el sistema de reescritura nunca detectará ni procesará reglas circulares.

Es necesario tener permiso de definición de reglas en una clase para poder definir una regla en él. Se debe utilizar el comando **GRANT** y **REVOKE** para modificar estos permisos.

El objeto en una regla SQL no puede ser una referencia a un arreglo y no puede tener parámetros.

Aparte del campo "oid", los atributos del sistema no pueden ser referenciados en ningún lugar de la regla. Entre otras cosas esto significa que las funciones de instancias (por ejemplo, foo(emp) donde emp es una clase) no pueden ser llamadas en ningún lugar dentro de una regla.

El sistema almacena el texto de la regla y los planes de consulta como atributos de texto. Esto implica que la creación de reglas puede fallar si la regla más sus varias internas representaciones exceden algún valor que es del orden de una página.

III. Preparación del ambiente de trabajo

Para poder aplicar los conceptos descritos en este laboratorio es necesario tener una base de datos en la cual aplicar las restricciones que requiere el proyecto de trabajo.

Creación de tablas

Las tablas que en esta sección encuentra se crearon aplicando las reglas de conversión del modelo E-R al relacional al diagrama E-R de la sección 1. Este laboratorio no intenta explicar esas reglas.

Esquemas para el diagrama E-R de la Universidad ACME:

Los nombres de los campos en algunos casos fueron cambiados, con respecto del diagrama E-R, por motivos de tamaños del nombre, sin embargo los conceptos son los mismos.

CuentaCheques(ncuenta, saldo, banco)
Cheque(ncuenta, cns, total, fecha);
Tabulador(idtab, importehora);
Profesores(idprofe, idtab, nombre, maximo, sueldo);
CursosEspeciales(idcurso, idprofe,cns,fini,ffin);
CursosExtracurricular(idextra, decextra, nhorascurso);
Materias(nmat, des, horacurso);
CEVerano(idcurso, nmat);
CEExtracurricula(idcurso, idextra);
Alumnos(matricula, nombre);
Bimestre(matricula, periodo, nmat, calificacion, faltas);
Faltas(matricula, fecha);
Concepto (idconcepto, desconcepto);
Recibo(folio,matricula, fecharec, totalrec);
DetalleRecibo(folio, cns, idconcepto, subtotal);
CalendarioEscolar(fecha, motivo);

Tablas para el diagrama E-R de la Universidad ACME:

Los siguientes comandos de creación de tablas, inserción de datos y creación de privilegios deben ser ejecutados usando el usuarios postgres (el usuario por omisión) y se debe de cambiar de usuario hasta que explícitamente se le indique. Si durante el proceso de instalación

de PostgreSQL, no ha creado un acceso directo para el programa psql, lo puede encontrar en la siguiente dirección: C:\Program Files\PostgreSQL\9.3\bin y construya el acceso directo. Ejecute el programa PSQL del postgreSQL (Shell de postgreSQL) y "copie" cada uno de los siguientes comandos y "péguelo" en el psql.

```
-- Creando la base de datos UACME
create database uacme;

-- Cambiarse de la BD por omisión a la ACME (en PSQL)
\c UACME

--Creación de las tablas
create table CuentaCheques(
ncuenta int,
saldo numeric(7,2),
banco varchar,
primary key(ncuenta)
);

create table Cheque(
ncuenta int,
cns int,
total numeric(10,2),
fecha date,
foreign key (ncuenta) references CuentaCheques,
primary key(ncuenta, cns)
);
create table Tabulador(
idtab int,
importehora varchar,
primary key(idtab)
);

create table Profesores(
idprofe int,
idtab int,
nombre varchar,
```

```
maximo varchar,
sueldo float, foreign key (idtab) references Tabulador,
primary key(idprofe)
);

create table CursosEspeciales(
idcurso int,
idprofe int,
fini varchar,
ffin varchar,
ncuenta int,
cns int,
foreign key(idprofe) references Profesores,
foreign key(ncuenta, cns) references Cheque,
primary key (idcurso)
);

create table CursosExtracurricular(
idextra int primary key,
decextra text,
nhorascurso int
);

create table Materias(
nmat int,
des varchar,
horacurso int,
primary key(nmat)
);

create table CEVerano(
idcurso int primary key,
nmat int,
foreign key(nmat) references Materias
);

create table CEExtracurricula(
idcurso int primary key,
```

```
idextra int,
foreign key (idextra) references CursosExtracurricular
);

create table Alumnos(
matricula int,
nombre varchar,
primary key(matricula)
);

create table Bimestre(
periodo int,
matricula int,
nmat int,
calificacion int,
Faltas float,
foreign key(nmat) references Materias,
foreign key(matricula) references Alumnos,
primary key(matricula, periodo)
);

create table Faltas(
fecha varchar,
matricula int,
foreign key(matricula) references Alumnos,
primary key(matricula, fecha)
);

create table Concepto(
idconcepto int,
desconcepto varchar,
primary key(idconcepto)
);

create table Recibo(
folio int,
matricula int,
fecharec varchar,
```

```
totalrec float,
foreign key (matricula) references Alumnos,
primary key(folio)
);

create table DetalleRecibo(
cns int,
idconcepto int,
folio int,
subtotal float,
foreign key(idconcepto) references Concepto,
foreign key(folio) references Recibo,
primary key(folio, cns)
);

create table CalendarioEscolar(
fecha varchar primary key,
motivo varchar
);
```

Inserción de datos para algunas tablas recién construidas. Los datos insertados sólo sirven para demostrar el funcionamiento de los privilegios de acceso, queda en manos del usuario insertar datos en el resto de las tablas para demostrar que las reglas de acceso son funcionales para cada usuario.

```
insert into CuentaCheques values(1,700,'HSBC');
insert into CuentaCheques values(2,9000,'HSBC');
insert into CuentaCheques values(3,60,'HSBC');
insert into CuentaCheques values(4,10,'HSBC');
insert into CuentaCheques values(5,1000,'HSBC');
insert into CuentaCheques values(6,200,'HSBC');

insert into Cheque values(1,10,200,'2008-02-01');
insert into Cheque values(2,10,575.20,'2008-02-01');
insert into Cheque values(2,20,20,'2008-02-01');
insert into Cheque values(3,10,600,'2007-02-01');
insert into Cheque values(4,10,800,'2007-02-01');
```

```
insert into Cheque values(5,10,100,'2007-02-01');
insert into Cheque values(6,10,300,'2007-02-01');

insert into Tabulador values(10,100);
insert into Tabulador values(20,200);
insert into Tabulador values(30,300);
insert into Tabulador values(40,400);
insert into Tabulador values(50,500);
insert into Tabulador values(60,600);
insert into Tabulador values(70,700);
insert into Profesores values(1,40,'Roberto','Maestr
ia',15000);
insert into Profesores values(2,70,'Carlos','Doctora
do',25000);
insert into Profesores values(3,20,'Luis','Licenciatura'
,6000);
insert into Profesores values(4,30,'Yunuan','Maestr
ia',12000);
insert into Profesores values(5,10,'Julio','Licenciatu
ra',4500);
insert into Profesores values(6,20,'Samuel','Licenciatu
ra',5500);

insert into CursosEspeciales
values(1,1,1,20070204,20050204);
insert into CursosEspeciales
values(2,2,2,20070204,20050204);
insert into CursosEspeciales
values(3,3,3,20070204,20050204);
insert into CursosEspeciales
values(4,4,4,20070204,20050204);
insert into CursosEspeciales
values(5,5,5,20070204,20050204);

insert into CursosExtracurricular values(1,'admin',204);
insert into CursosExtracurricular values(2,'diseño',204);
insert into CursosExtracurricular values(3,'bdd',204);
insert into CursosExtracurricular values(4,'java',204);
```

```
insert into Materias values(1,'admin bdd',204);
insert into Materias values(2,'redes',204);
insert into Materias values(3,'redes 2',204);
insert into Materias values(4,'admin bdd',204);
```

IV. Problemática a resolver

En esta sección ponemos manos a la obra resolviendo el problema plateado inicialmente. En este momento ya tenemos la base de datos creada, hemos insertado datos en algunas tablas y es hora de implementar los conceptos que hemos estudiado en la sección conceptos básicos. Los problemas aquí planteados se han resuelto, por lo que debe de "copiar" el comando SQL y "pegarlo" en el psql.

Administración de usuarios
Resolviendo el problema de la administración de UACME. Se han identificado dos grupos de usuarios en este sistema: Administración y Escolar. Dentro de cada uno de estos grupos encontramos a los usuarios, los que se clasifican así:

Administración:
Jefe del departamento: C. P. Ávila
Auxiliar de Administración: Sr. Cancino

Escolar:
Jefe del departamento: Lic. Barroso
Auxiliar Escolar: Sra. Tirado
Auxiliar Escolar: Sra. Martínez
Auxiliar Escolar: Sra. Aquino
Auxiliar Escolar: Sra. Ramos

Creación de los grupos de usuarios
Se han elegido los nombres de grupos Admin para la DA y Escolar para el DCE.

```
-- Creando los grupos de usuarios
```



```
create group admin;
create group escolar;
```

Creación de usuarios del grupo Administración
Se les ha asignado como password el nombre del grupo, pero esto se hace por facilidad, en la práctica cada usuario debe elegir un password personalizado.

```
-- Creando los usuarios del grupo administrativo
create user avila with password 'admin' in group admin;
create user cancino with password 'admin' in group admin;
```

Creación de usuarios del grupo Escolar
Se les ha asignado como password el nombre del grupo, pero esto se hace por facilidad, en la práctica cada usuario debe elegir un password personalizado.

```
-- Creando los usuarios del grupo escolar
create user barroso with password 'escolar' in group escolar;
create user tirado with password 'escolar' in group escolar;
create user martinez with password 'escolar' in group escolar;
create user aquino with password 'escolar' in group escolar;
create user ramos with password 'escolar' in group escolar;
```

Otorgamiento de privilegios de acceso sobre las tablas del sistema
Para otorgar privilegios de acceso la junta directiva de UACME ha autorizado el acceso sobre las siguientes tablas para cada departamento:

Tablas a las que se le permite el acceso al grupo Administración
CuentaCheques(ncuenta, saldo, banco)

Cheque(ncuenta, cns, total, fecha);
Tabulador(idtab, importehora);
Profesores(idprofe, idtab, nombre, maximo, sueldo);
Concepto(idconcepto,desconcepto);
Recibo(folio,matricula,fecharec,totalrec);
DetalleRecibo(folio,cns,idconcepto,subtotal);

Como casos especiales este departamento podrá acceder a consultar las tablas de Cursos Especiales, Cursos Especiales Verano, Cursos Especiales Extracurriculares, Cursos Extracurriculares y Materias. Explícitamente no se les permite modificar ningún campo o registro.

Tablas a las que se le permite el acceso al grupo Escolar

CursosEspeciales(idcurso, idprofe,cns,fini,ffin);
CursosExtracurricular(idextra, decextra, nhorascurso);
Materias(nmat, des, horacurso);
CEVerano(idcurso, nmat);
CEExtracurricula(idcurso,idextra);
Alumnos(matricula, nombre);
Bimestre(matricula,periodo,nmat,calificacion,faltas);
Faltas(matricula, fecha);
CalendarioEscolar(fecha, motivo);

Caso especial de profesores para el grupo Escolar

La forma en que evitaremos que el grupo escolar vea el sueldo del docente es creando una vista y asignando privilegios por separado.

```
-- Creando la vista, note que no se proyecta el campo
sueldo, por lo tanto se esconde de la mirada de los
usuarios del grupo escolar.

CREATE VIEW VistaProfesoresEscolar AS SELECT
idprofe,idtab,nombre,maximo FROM Profesores;
```

Otorgando los privilegios en la BD al grupo Administración en el SQL

```
-- Todos los privilegios para las tablas del sistema
administrativo
```

```
GRANT all ON table Cheque to group admin;
GRANT all ON table CuentaCheques to group admin;
GRANT all ON table Tabulador to group admin;
GRANT all ON table Concepto to group admin;
GRANT all ON table Recibo to group admin;
GRANT all ON table DetalleRecibo to group admin;

-- Solo puede consultar y actualizar la tabla de
profesores a los usuarios del grupo admin.
GRANT select, update ON table Profesores to group
admin;

-- Solo se permite consultar estas tablas a los
usuarios del grupo admin.
GRANT select ON table CursosExtracurricular to group
admin;
GRANT select ON table CEVerano to group admin;
GRANT select ON table CEExtracurricula to group admin;
GRANT select ON table CursosEspeciales to group admin;
```

Otorgando los privilegios en la BD al grupo Escolar en el SQL

```
-- Otorgando privilegios sobre las tablas del sistema
escolar.
GRANT all ON table Materias to group escolar;
GRANT all ON table CursosExtracurricular to group escolar;
GRANT all ON table CEVerano to group escolar;
GRANT all ON table CEExtracurricula to group escolar;
GRANT all ON table CursosEspeciales to group escolar;
GRANT all ON table Alumnos to group escolar;
GRANT all ON table Faltas to group escolar;
GRANT all ON table Bimestre to group escolar;
GRANT all ON table CalendarioEscolar to group escolar;
-- Otorgando permisos sobre la vista de profesores
GRANT all ON table VistaProfesoresEscolar to group escolar;
```

Prueba de los privilegios asignados a los usuarios

Para conectarse a la base de datos con un usuario diferente a postgres (ver Figura 2) busque en el directorio “C:\Program Files\PostgreSQL\9.3\bin” el comando psql e indíquele a cual base de datos se desea conectar (*para nuestro caso -d uacme*) y el usuario que va a usar (*-U barroso*), una vez hecho esto PostgreSQL le solicita el password para el usuario barroso y en caso de ser correcto se despliega el shell de la BD. El ejemplo está desarrollado sobre la plataforma Windows. Intente conectarse con los diferentes usuarios que ha creado previamente.

```
C:\Program Files\PostgreSQL\9.3\bin>psql -d uacme -U barroso
Password for user barroso:
Welcome to psql 9.3.4, the PostgreSQL interactive terminal.
Type: \copyright for distribution terms
\h for help with SQL commands
\? for help with psql commands
\g or terminate with semicolon to execute query
\q to quit
uacme=>
```

Figura 2. Ejemplo de conexión con usuario diferente a postgres.

• Conectado a la BD uacme con el usuario barroso

Intente los siguientes comandos:

```
Select * from Profesores;
Select * from VistaProfesoresEscolar;
Select * from CuentaCheques;
Select * from Cheque;
insert into Profesores values(7, 20, ‘Jesus’,
‘Licenciatura’, 25000 );
insert into VistaProfesoresEscolar values(7, 20, ‘Jesus’,
‘Licenciatura’, 25000 );
```

Esto nos obliga a atacar el problema con una nueva perspectiva, la creación de reglas, **usando la cuenta del usuario postgres** ejecute el siguiente comando:

```
CREATE or REPLACE RULE insertar_profesores AS ON insert
TO VistaProfesoresEscolar
DO INSTEAD insert into profesores values (
new.idprofe,
new.idtab,
new.nombre,
new.maximo, 0);
```

Este problema aparece debido a que no se permite insertar registros sobre vistas, entonces lo que tenemos que hacer es redirigir la operación a la tabla profesores usando una regla (la cual es una especie de interrupción) que se dispara cada vez que se intenta ejecutar una inserción de datos sobre la tabla.

Con la cuenta del usuario Barroso intente nuevamente la ejecución del comando:

```
insert into VistaProfesoresEscolar values(7, 20, 'Jesus',
'Licenciatura');
```

- **Conectado a la BD uacme con el usuario avila**, intente los siguientes comandos:

```
Select * from Profesores;
Select * from VistaProfesoresEscolar;
Select * from CuentaCheques;
Select * from Cheque;
update Profesores set sueldo = 6000 where idprofe = 7;
insert into Profesores values(8, 30, 'Salvador',
'Maestria', 20000 );
insert into VistaProfesoresEscolar values(8, 30,
'Salvador', 'Maestria');
```

V. Trabajo adicional

Los siguientes problemas no están resueltos, por lo que es necesario aplicar su experiencia adquirida para resolver estos.

1. Usando a los distintos usuarios verifique que se le permitan efectuar los movimientos acordes al privilegio de acceso asignado sobre cada una de las tablas.

2. Construya la regla faltante para el grupo escolar sobre la vista *VistaProfesoresEscolar*, construya la regla para cuando el usuario barroso desea eliminar el registro del docente Luis.

3. Agregue restricciones adicionales. Use la siguiente política: El encargado de capturar los cursos especiales será el departamento administrativo (cualquier usuario), pero quién asignará el docente será el departamento escolar. Construya la vista y la regla que va a reglamentar esta inserción de datos.

    - La inserción de datos a la tabla CursosEspeciales y las tablas especializadas la efectuará el DA, con el campo profesor en nulos o referenciando a un registro especial (por ejemplo, un profesor no válido) de la tabla Profesores.

    - La modificación de las tablas CursosEspeciales para asignar al Profesor la efectuará el DCE.

4. Finalmente, explique las causas por las que las instrucciones para los usuarios: barroso y ávila, marcan errores o funcionan adecuadamente.